From concatenated codes to graph codes

Justesen, Jørn; Høholdt, Tom

Published in:
IEEE Information Theory Workshop, San Antonio

Link to article, DOI:
10.1109/ITW.2004.1405266

Publication date:
2004

Document Version
Publisher's PDF, also known as Version of record

From Concatenated Codes to Graph Codes

Jorn Justesen
COM, Technical University of Denmark
DK-2800 Kgs Lyngby, Denmark
e-mail: jju@com.dtu.dk

Tom Hoholdt
Department of Mathematics
Technical University of Denmark
DK-2800 Kgs Lyngby, Denmark
e-mail: T.Hoeholdt@mat.dtu.dk

Abstract — **We consider codes based on simple bipartite expander graphs. These codes may be seen as the first step leading from product type concatenated codes to more complex graph codes. We emphasize constructions of specific codes of realistic lengths, and study the details of decoding by message passing in trees.**

I. Introduction

The codes considered in this paper are a specific case of the codes based on bipartite expander graphs described in [1, 2]. By specializing the analysis to a class of graphs with well-understood properties we are able to obtain more details. Even though this is only the first step in the generalization that leads from product type concatenated codes to more complex graph codes, the codes are already about as long as it would be practical for any implementation.

II. A Class of Bipartite Graphs from Projective Planes

Let M be a cyclic incidence matrix for a projective plane of order n-1 with S lines and points.

$$S = n(n-1)+1$$

Thus each row has n 1s, the largest eigenvalue is n and the corresponding eigenvector is the all-ones vector. The scalar product of any two different rows is one, and all remaining eigenvalues have modulus $\lambda = \sqrt{n-1}$. A bipartite graph consisting of two sets of S nodes of order n is described by the incidence matrix

$$A = \begin{bmatrix} 0 & M \\ M' & 0 \end{bmatrix}$$

This matrix may be seen as a simple expander graph: Starting from a node in the right set, n nodes in the left set can be reached in one transition, and the remaining nodes in the right set can be reached from these nodes. The eigenvalues of A are $\pm n$ and $\pm\lambda$ (all real since A is symmetric).

The graph can be used to define a code by associating a symbol with each branch and letting all branches that meet in a node satisfy the parity checks of an (n, k, d) code. Thus the length of the code is

$$N = Sn$$

If the rate of the code associated with the nodes is r, the total rate is

$$R \geq 2r - 1$$

To obtain a lower bound on the minimum distance, we want to determine the smallest size of sets of nodes in each of the two parts of the graph, s, such that the subgraph consisting of these nodes and the branches connecting them has degree at least d. Clearly in this case sd is a lower bound on the minimum distance.

We give details only of the case where d is the same for the two sets of codes. In the asymptotic analysis the rates and distances are different, and a more general result is needed.

III. A Lower Bound on s

In [1, 2] the size of the sets considered in Section 2 was bounded in terms of λ. Our argument will follow the proof from [2] closely, but provide some additional detail. Thus X will be a vector of length $2S$ with coordinates 1 in s places indicating the positions in each of the two sets, and Y is the corresponding balanced vector

$$Y = X - \frac{|X|}{2S}U$$

where U is a vector of 1s. Thus Y may be expressed as a sum of those eigenvectors of A that have eigenvalue $\pm\lambda$. Following the steps of the proof in [2], we get

$$d \leq ns/S + \lambda(1 - s/S)$$

with equality if Y is an eigenvector with eigenvalue λ.

Since Y has s coordinates with value $1-s/S$ while the remaining are $-s/S$, and we may assume that each of the s nodes in the right subset is connected to exactly d nodes in the left subset (and thus n-d other nodes in the left set), we can find the coordinates in the subsets of AY to be

$$(1 - s/S)(Sd - ns)/(S - s)$$

On the average, the remaining coordinates are scaled by the same factor. Thus for Y to be an eigenvector we must have

$$\lambda = (Sd - ns)/(S - s)$$

and in general the right side is at most equal to λ. From earlier relations between the parameters we get

$$s \geq (d - \lambda)(\lambda^2 + \lambda + 1)$$

Clearly equality is possible only for integer values of λ. For small d, the bound is useless, and we can have s=d(d-1)+1. This follows from the expansion property of the graph:

0-7803-8720-1/04/$20.00 ©2004 IEEE

Consider a node on the right with at least one nonzero symbol. There has to be at least d nonzero branches leading to left nodes. These are again connected to $d(d-1)$ distinct right nodes. In the case when the projective plane used in the construction contains a smaller projective plane over a subfield, we can have

$$d = \lambda + 1$$

Thus the minimum distance is always lower bounded by

$$d((d-1)d+1) = d(d^2 - d + 1)$$

If the two codes have the same length, but different minimum distances, we can apply the same argument starting from either side of the graph. Thus the minimum distance is lower bounded by

$$\max\{d_1((d_2-1)d_1+1), d_2((d_1-1)d_2+1)\}$$

These two terms can be interpreted as minimum weights of codewords in tree codes rooted on the right or left side of the graph. Thus in a decoding based on tree codes we would use the version that gives the larger term, and this will equal the estimate on the minimum distance of the code, d'.

It follows immediately that for any error pattern of weight less than $d'/2$, the branches directly connected to the root of the tree are correctly decoded. However, peripheral branches may not be correct, and in that case the estimated total number of errors is not necessarily correct. By combining the decoding results for all tree codes, we will of course get the correct codeword.

Example 1: For S=21 and λ=2 we get d=3 as the smallest case, and the subgraph is the incidence matrix of the projective plane with s=7. For d=4 we get the complementary sets, and in both cases Y is an eigenvector of A.

The lower bound on s indicates that in order to get a strong bound on the minimum distance of the graph code, the minimum distances of the component codes have to exceed the square root of the length. Since the codes (or at least one code) also must have moderately high rate, the component codes cannot be short. In case of binary codes the bound applies for $n \geq 256$. It is easier to get a useful construction with RS component codes, and we shall discuss such codes in the following section.

IV. Generalized Concatenated Codes

The same type of bipartite graphs can be used with q-ary connections (q a power of 2) and RS component codes. To get an asymptotic result, we assume that codes in the right subset of nodes have a moderate rate r_1 and codes in the left subset have rate r_2 close to 1. This will lead to a Forney type decoding of the concatenated code. The codes on the right side can be interpreted as binary codes, and their properties can be made to approach that of a random ensemble by a suitable randomization of the mapping of the symbol field.

The analysis of the weight of these binary codes (C. Thommesen [3]) shows that the minimum distance is on the GV bound for binary codes of rate r_1, but also that the minimum distance is reached only for codewords of maximal q-ary weight. Words with a smaller number of nonzero symbols have larger minimum weighs. As pointed out by Barg [4] this observation leads to a lower bound on the minimum weight of the concatenated code that exceeds the product bound.

For moderate values of n, there are cases where the binary images of RS codes are known to have good minimum distances. In such cases there are often many low weight codewords, and the restriction on the number of nonzero q-ary symbols does not necessarily lead to an increased minimum weight. In this case r_2 should also be chosen as a lower value.

V. Outline of a Decoding Algorithm

The following algorithm is designed to make use of the graph structure of the code to decode beyond half the minimum distance. The algebraic properties are used to reduce the complexity and to allow the performance of at least some of the steps to be analyzed. The decoding consists of 5 stages:

- Decode the binary images of the right side codes to a distance of $(1-\varepsilon)nH^{-1}(1-\eta)$, i.e. close to the Hamming bound for binary codes of rate r_1. It is known that the number of words within this distance is upper bounded by a constant that is a function of ε, but does not depend on the length [5]. Clearly the average number of words is only slightly greater than 1. For each $F(q)$ symbol in a given position, propagate a message along the branch in the graph indicating the minimum number of binary errors corrected in the first stage of decoding. If no word was found with the symbol in question, the value of the Hamming bound is used as an estimate.
- Using these messages, decode the left codes as RS codes. Use information set decoding [5] (or another suitable approach, maybe Koetter-Vardy decoding [6]) to find codewords within the small lists of symbols provided by the first stage and an additional small number of errors. The advantage at this stage over standard concatenated codes is that there is a large number of codes, but that they are smaller.
- Pass the result to the right side, and consider these codes as RS codes. Each code on the right side is now the root of a tree code consisting of all codes on the right side, a small subset of codes on the left side, and all symbols in the total code. When decoded, it provides an estimate of the total number of binary errors corrected (since this number was propagated as part of the messages from each stage). Decoding by message passing is ML for each tree code, and thus the actual number of errors cannot be smaller than the maximum estimated by the S tree codes.
- Choose a set of n nodes on the right among the S sets that are connected to a common node on the left. The set with the largest number of estimated errors is preferred, but in addition the decoded symbols on the branches connecting to the common left node must be a codeword. Let T be the largest estimated number of errors among the tree codes in this set. Propagate the decoding decisions consistent with at most T errors to all remaining nodes on the left.
- Using the symbols from third decoding stage, choose the corresponding codeword in the left codes such that the decisions are consistent for all symbols and the total number of errors is T. If no result is found, increase T and repeat the process.

As in the original Forney codes [7], the construction asymptotically reaches capacity, although the error exponent is reduced. This is done by increasing the length

of the codes and letting r_2 approach 1. However, in the codes considered here, the length of the total code increases quickly, while the component codes remain moderate. The set of tree codes used in the decoding includes all component codes, and thus a decoding result is a codeword in the total code. If a codeword is found, it is ML since message passing in tree codes is ML and all error patterns of lower weight have been eliminated. However, the error rate cannot exceed the Hamming bound for r_1, and if T has to be increased much from its initial value, the last step in the decoding becomes exponentially complex. For codes of moderate length and minimum distance in the order d^3, the algorithm corrects errors of weight less than half the minimum distance, but also many error patterns of higher weight.

VI. Examples

Example 2: We consider the details of the construction and the decoding performance for a small example based on (5,3,3) RS codes over $F(4)$ and a 42 node bipartite graph derived from the 21 point projective plane constructed over $F(4)$. The dimension of the code depends on the assignment of code positions to branches of the graph. Since M is chosen to be cyclic, we can assign positions 1, 2, .. , n to the branches that correspond to the 1s in the first row of M taken in the order in which they appear in that row. Similarly the positions of the component codes associated with the following right nodes in the graph are found as cyclic shifts of the first assignment. The codes of the left nodes can be assigned using the same labels, and thus a given branch has the same position in both codes. In $F(4)$ the parameters of the code are (N,K,D)=(105, 23, 21). In $F(2)$ (N, K) = (210, 46).

The decoding is based on comparing the results of decoding 21 tree codes which include all symbols of the code (80 of these as leaves). Each tree consists of an RS code as the root. This code is connected by 5 symbols to other RS codes, and these are connected to 20 binary codes. A codeword of minimum weight with a nonzero root has three nonzero symbols at this level, two additional nonzero symbols in each of the next level RS codes, and two additional nonzero symbols in each of 6 binary codes. The $F(4)$ minimum weight is 21 and equal to the minimum weight of the code.

The binary image of the component code is a (10, 6, 3) shortened Hamming code. The message associated with a branch connecting the code to the RS code below is the minimum number of bit errors assuming each of the 4 possible values of the branch symbol. In general, messages in the decoding of the tree code are vectors of estimated numbers of bit errors in the subtrees above a given branch conditioned on the symbol values on the branch. We could find these messages exactly by decoding each of the shortened (8, 4) codes, but we assume a simplified approach where each code is decoded only once, and the message is (0, 3, 3, 3) for a correct codeword, (1, 2, 2, 2) when one error has been corrected, and (2, 2, 2, 2) when the error is in one of the remaining 4 cosets. For a given error probability we can calculate the probability distribution of the message vectors, and in a particular tree code, the messages from the 20 binary codes are independent.

In the decoding of the RS codes in the second stage, we fix the symbol on the output branch to a given value, and then find the codeword that gives the smallest sum of the weights in the 4 inputs. Again we can simplify this step by performing a list-in-list-out decoding to obtain the codewords of smallest weight and use a lower bound for the remaining outputs.

The RS codes of the third stage are the same as the codes of the first stage, but they are now decoded in the bigger field using the messages generated by the second decoding stage. The sum of the messages from the previous state is a lower bound on the number of errors in the tree. The result of the decoding is also a preliminary decision on the 5 symbols.

Each left node is connected to a set of 5 right codes. Select a corresponding set of 5 RS codes in the last stage such that they have the largest possible number of estimated bit errors (since taking the highest value of the lower bound gives the best approximation to the actual error pattern). However, the decisions on the symbols connecting to the common left node must also be a codeword.

The messages are not useful unless most of the estimates in the first stage are correct. Thus we can expect to correct 20 errors, but not a much larger number. A typical distribution is for the 20 codes in the first stage to have 7, 8, 4, 1 codes with 0, 1, 2, and >2 errors. Thus of the 20 messages, 3 or 4 have a higher weight for correct symbol value than for some other value, and since the RS codes can correct an error, they will decide in favor of the correct word, possibly with a single exception. Thus in stage 3, the decision will be correct, and the estimated number of bit errors will be 2-4 below the actual number. The actual error pattern is easily corrected by propagating the decisions from the root. In a worst case situation, the 20 errors give rise to erroneous changes of one position in 10 codes in the first stage. However, in the next stage each RS code on the average has 2 symbols with weights 2 to 1 in favor of an incorrect value, but this correction will be preferred to changing a symbol from an error-free code where the weights are 0 to 3. Clearly a relatively small number of error patterns will be close enough to low weight codewords to cause decoding errors.

Example 3: For the stronger bound on the minimum distance to apply, we can use RS codes of length 64. At this length there is still little room for the codes to have different rates, so we let both component codes be (n, k)=(64, 40). Thus the length of the binary code is about $N \approx 6 \cdot 2^{18} \approx 1.5 \cdot 10^6$, and $D \approx 64 \cdot 16 \cdot 40 \approx 41{,}000$. Even though the minimum distance is not very good, the error probability would be negligible. The performance is limited by the rate of the inner binary code, which has to be correctly decoded in most cases. For comparison, a similar standard concatenated code could use the same inner code and 20 interleaved outer RS codes over $F(2^{12})$. The graph code would have a slight advantage in decoding complexity by using a smaller field, but the overall rate is ¼ compared to about 3/8 for the concatenated code with the same outer code rate.

VII. Conclusion

Asymptotic analysis of concatenated codes based on expander graphs indicates that they could have performances similar to standard concatenated codes with a lower decoding complexity. For codes of moderate lengths, the restrictions on the rates of the component codes lead to a significant loss in the overall rate. To obtain codes that have advantages over standard concatenated codes, the lengths would have to be in the range 10^7 to 10^9, which might still be realistic.

REFERENCES

[1] A. Barg, G. Zemor, "Error exponents of expander codes," IEEE Trans. Inform. Theory, vol. IT-48, pp. 1725-29, June 2002.

[2] G. Zemor, "On expander codese," IEEE Trans. Inform. Theory, vol. IT-47, pp. 835-837, February 2001.

[3] C. Thommesen, "The existence of binary linear concatenated codes with Reed-Solomon outer codes which asymptotically meet the Gilbert-Varshamov bound ", IEEE Trans. Inform. Theory, vol. IT-29, pp. 850-853, November, 1983.

[4] A. Barg, G. Zemor, "Concatenated codes," manuscript, 2003.

[5] V. V. Zyablov, M. S. Pinsker,"List cascade decoding",Probl. Pered. Inform., vol. 17, no. 4, pp. 29 - 34, 1981. (English translation: Probl. Inform. Transm., pp. 236-240, 1982)

[6] R. Koetter & A. Vardy,"Algebraic soft-decision decoding of Reed Solomon codes ", IEEE Trans. Inform. Theory, vol. IT-49, pp. 2809 - 2825, November, 2003.

[7] G.D. Forney, Jr., Concatenated Codes, MIT Press, 1966.